Abstract - An expert system GIFTS (a Guide to Intelligent Fusion Technology Selection) developed to aid sensor fusion system design, was presented at Fusion 98 as an on-going project with additional support tools under development. In this paper, a simulation tool, FUSE, that exercises a decisions in - decision out (DEI-DEO) fusion model to estimate the benefits (utility) along a sequence of multiple observations, is presented. This can be employed either independently or as one of the tools supporting GIFTS. FUSE permits the simulation of different Boolean fusion logic functions in the context of sensor suites with two independent sensors. The inputs to FUSE are the sensor performance characteristics in terms of the probabilities of correct and incorrect decisions for target and decoy classes along with parameters that define the fusion logic and duration. The outputs of the system are the fused system performance expressed in terms of probabilities of correct-, incorrect- and non- decisions over the specified range of observations. Whenever any of these input parameters are altered, FUSE responds instantaneously by updating the fused system performance. In order to further aid the user in the comparison of the different fusion logic alternatives and to assess the benefits of temporal fusion through multiple independent observations, FUSE provides several graphic visualization options.

Key Words: decision fusion, fusion benefits, fusion logic, temporal fusion

1. Introduction

A common question that arises, especially from outside the sensor fusion community, is why fuse the sensors at all. An often heard comment is: “Why should I fuse my two sensors when sensor number one has superior performance? I will just be degrading its performance by mixing in less reliable information.” Of course fusing sensors can be beneficial, but it is often hard to convey this message without showing hard data to the skeptic.

Even assuming that one is convinced of the advantages of pursuing fusion, a second question is often how the fusion should be accomplished. This is the challenge of determining what to fuse, when to fuse [1], and how to fuse. There is abundant literature offering different methods of accomplishing fusion [2,3], but few universal rules to follow. Instead, each scenario has been individually analyzed and all methods have to be considered in the appropriate context.

FUSE (Fusion Utility Sequence Estimator) is designed to address both of these questions, albeit in a limited fashion. To help with the first problem (addressing the utility of fusion), FUSE can be used as a stand-alone fusion simulator. A user simply inputs appropriate values for the individual sensor characteristics through the user interface and fused decision probabilities are immediately updated. Hence, the advantages of fusion can be instantaneously gleaned. To further aid in examining the fusion benefits, graphical representations can be displayed.

FUSE, when used in conjunction with GIFTS (a Guide to Intelligent Fusion Technology Selection) [4], addresses the larger problem as well. GIFTS guides a user through an interactive query session that defines a fusion system architecture that is appropriate to the problem environment under consideration. Included within the GIFTS architecture, are several support tools that provide assistance to the designer in developing and assessing the detailed fusion system design corresponding to the chosen architecture. FUSE is an additional assessment tool that can be included in the GIFTS architecture or can be operated as a stand-alone fusion simulator. FUSE employs a decisions in - decision out (DEI-DEO) fusion model to estimate the benefits along a sequence of multiple observations using a two-sensor suite under AND and OR boolean logic.

In this paper, the initial version of FUSE will be introduced. In Section 2, an overview of the basic
fusion concepts underlying the estimation techniques is presented along with a summary of the GIFTS architecture. Section 3 is a description of the FUSE simulation and section 4 presents how FUSE can be used as a realistic analysis tool. Section 5 offers some closing comments and outlines the scope for further development.

2. Background

This section contains the basics of the methods by which FUSE estimates fusion benefits and the GIFTS architecture. For more details on the estimation techniques see [5]. Those interested in GIFTS should look up [4].

2.1 Fused Probability Estimation

There are two basic fusion strategies that are used in FUSE. The two strategies are OR and AND boolean logic. Both strategies operate in an environment where two sensors are operating in parallel, have a provision for multiple looks, and have a non-decision option as well as the normal binary decisions. OR logic fuses decisions by making a binary decision if the two sensors are not contradictory and a non-decision otherwise. AND logic, on the other hand, requires that the two sensors make concurring decisions to obtain a binary decision and a non-decision otherwise.

Let $c_{ij}, w_{ij},$ and $u_{ij}$ correspondingly represent the probabilities of correct, incorrect, and non-decision of objects $j \in \{ \text{Target (T)}, \text{Decoy (D)} \}$ by the sensors $i = \{1, 2\}$, where both sensors are deemed independent. Similarly, $p_{fj}^k, q_{fj}^k,$ and $r_{fj}^k$ represent the fused probabilities of correct, incorrect, and non-decision of object $j$ after the $k^{th}$ fusion attempt under logic $f = \{ \text{OR (o)}, \text{AND (a)} \}$. Using these definitions we note that

$$c_{ij} + w_{ij} + u_{ij} = 1.$$  \hfill (1)

It can then be shown for OR logic that

$$p_{oj}^1 = c_{1j}c_{2j} + c_{1j}w_{2j} + u_{1j}c_{2j}$$  \hfill (2)

$$q_{oj}^1 = w_{1j}w_{2j} + w_{1j}u_{2j} + u_{1j}w_{2j}$$  \hfill (3)

$$r_{oj}^1 = u_{1j}u_{2j} + c_{1j}w_{2j} + w_{1j}c_{2j}.$$  \hfill (4)

Similarly, the following equations can be developed for AND logic.

$$p_{oj}^1 = c_{1j}c_{2j}$$  \hfill (5)

$$q_{oj}^1 = w_{1j}w_{2j}$$  \hfill (6)

$$r_{oj}^1 = u_{1j}u_{2j} + c_{1j}w_{2j} + w_{1j}c_{2j}.$$  \hfill (7)

The $k^{th}$ probabilities can thus be written as

$$P_{fj}^k = \sum_{i=1}^{k} p_{fj}^i = p_{fj}^i \frac{1 - [r_{fj}^i]^{k}}{1 - r_{fj}^i}$$  \hfill (8)

$$Q_{fj}^k = \sum_{i=1}^{k} q_{fj}^i = q_{fj}^i \frac{1 - [r_{fj}^i]^{k}}{1 - r_{fj}^i}$$  \hfill (9)

$$r_{fj}^k = 1 - P_{fj}^k - Q_{fj}^k$$  \hfill (10)

$$r_{fj}^1 = 1 - p_{fj}^1 - q_{fj}^1.$$  \hfill (11)

These eleven equations form the basis of the fusion benefit analysis in FUSE.

Three types of fusion benefits will be defined. These are with respect to the probability of correct decision, probability of incorrect decision, and both. A fusion benefit exists with respect to the probability of a correct decision when

$$P_{fj}^k > \max(c_{1j}, c_{2j}).$$  \hfill (12)

Similarly, the fusion benefit with respect to the probability of an incorrect decision would be

$$Q_{fj}^k < \min(w_{1j}, w_{2j}).$$  \hfill (13)

A joint fusion benefit is thus when both (12) and (13) simultaneously hold true.

2.2 GIFTS

GIFTS is currently composed of four modules. The primary component is the architecture selection process that determines the relevant architecture. The second piece is a reference database that can be used to help answer problem specific questions. The third part is an FEI-DEO fusion selector. It provides a means of choosing an implementation of FEI-DEO fusion. The final component is FUSE that is discussed in this paper to simulate DEI-DEO fusion.

Through the use of all the modules, GIFTS can provide aid to the fusion system architecture designer during the different phases of development. The user, who knows the specifics of a fusion problem, uses GIFTS to determine a proposed architecture. This is accomplished by responding to problem spe-
cific questions posed by the primary component of GIFTS. Once the proposed architecture has been created, the user will begin a problem specific refining process that will produce a fusion solution. During this time, the goal is to determine the optimal means of implementing the different fusion modes in the proposed architecture. (Of course the option of not utilizing a fusion mode in the proposed architecture is always available. It may be that even though fusion is practical in this mode, there is no reasonable means of implementing it in the user’s application, or a restraint outside the realm of GIFTS could be a limiting factor.)

It is at this point in the development process that the remaining modules of GIFTS will be useful. The reference tool can be used to provide sources of information on different fusion levels. The reference tool will provide a list of references that are related to the fusion modes in the proposed architecture. Thus, the reference tool makes use of the knowledge gained by the primary component. Similarly, if the user has not previously determined methods for performing FEI-DEO fusion or making local decisions, then the FEI-DEO fusion selector will be helpful. In this tool, the user is asked application specific questions to determine the most appropriate implementation method. The FUSE tool would be used to investigate DEI-DEO fusion as discussed in this paper.

3. The FUSE Tool

FUSE is currently implemented on a PC using Visual C++ [6]. It will thus run with no alterations on Windows 95, Windows 98, or Windows NT. The user can alter the sensor characteristics by choosing "Fusion Inputs" from the Fusion menu, or by clicking on the fusion characteristics button on the toolbar. Figure 1 shows the user interface with the Fusion menu activated. After "Fusion Inputs" has been chosen, the Data Definition Dialog Box (DDDB) will appear. It is through this dialog box that the fusion characteristics can be altered. Figure 2 is the default setting of the DDDB.

The DDDB consists of two group boxes labeled "Inputs" and "Fused Decision Probabilities", respectively along with a button label "OK."

The "Inputs" group is where the fusion characteristics are controlled. The inputs that can be altered are: decision probabilities for sensor 1, decision probabilities for sensor 2, the type of fusion logic, and the number of looks permitted for each sensor. Note that the user can have control over the correct and incorrect decision probabilities for both of the possible binary decisions (T and D). This allows for the maximum flexibility in the definition of a sensor. These probabilities can be entered by directly typing in the desired number of by using the adjacent sliders. It should also be pointed out that from equation (1), the four non-decision probabilities are uniquely defined by the inputs. The fusion logic is selected by a simple check box (checked for AND logic and unchecked for OR logic). The number of looks are entered by typing the appropriate integer in the box labeled "Number of Looks."

The "Fused Decision Probabilities" group is where the fused results, based on the above inputs, are displayed. The fused correct, incorrect, and non-decision probabilities are displayed for both the T and D binary decision.

The calculation of fused probabilities is only one aspect of FUSE. FUSE also provides a collection of visualization tools to help analyze the results. Each of these options are accessed through the “Fusion” menu. (Note that to activate the “Fusion” menu the input dialog cannot be open. Hence, if the dialog box is open then the “OK” button needs to be selected to exit the dialog box.) The graphical options are:

"Plot Target" - plots the fused correct, incorrect, and non-decision probabilities, for the T object, against the number of looks,

"Plot Decoy" - plots the fused correct, incorrect, and non-decision probabilities, for the D object, against the number of looks,

"Plot Target And/OR" - plots the fused correct probability, for the T object, under both AND and OR logic against the number of looks,

"Plot Decoy And/OR" - plots the fused correct probability, for the D object, under both AND and OR logic against the number of looks,

"Plot Target Fusion Benefits" - plots the fused correct and incorrect probabilities, for the T object, against the number of looks while shading the domain of fusion benefit for each and marking the domain of joint benefit, and

"Plot Decoy Fusion Benefits" - plots the fused correct and incorrect probabilities, for the D object, against the number of looks while shading the domain of fusion benefit for each and marking the domain of joint benefit.

The definition of a domain of joint fusion benefit is the intersection of the domains of correct and incorrect fusion benefit. The domain of correct (incorrect) fusion benefit is the domain where the correct (incorrect) fusion benefit exists. The correct and incorrect fusion benefit domain can be thought of as the domain bounded by a fused probability curve (with respect to the number of looks) and the best performance of a single sensor. As an example, if the probability of correct, incorrect, and non-decision for sen-
sor 1 are 71%, 16%, and 13% respectively and 45%, 15%, and 40% for sensor 2, then the best performance for a single sensor would be a correct-decision probability of 71%. Thus the fused correct probability curve and a horizontal line would bound the domain of fusion benefit for the probability of a correct decision at 71%.

4. FUSE Usage Illustration

For FUSE to be of practical value, one needs to be able to exercise it in a realistic scenario. It is meant to be a utility to assist a fusion system designer. To demonstrate its utility, consider the following scenario. A fusion system is being designed that employs two sources of decisions (or sensors) that are independent and capable of multiple looks. (An example of such a system would be a target acquisition system that employs an active X-band radar and a passive IR sensor.)

The goal is to balance the performance requirements of the system against the costs. Often this balance is obtained while using individual sensors that make decisions below system specifications and obtain decision probabilities that meet specifications through fusion. For example, the system in this scenario requires that the probability of correct and incorrect decisions for the target object are 95% and 1% respectively, while these probabilities are 90% and 5% for the decoy object. From a cost-benefit analysis, it was determined that each sensor will be manufactured to produce at best a 65% - 70% probability of a correct decision and 7% - 10% probability of incorrect decision. Also, the maximum number of looks desired should be between 5 and 8.

Initial values are first chosen in the analysis. In this case, sensor one has probabilities of correct (T), correct (D), incorrect (T) and incorrect (D) of 0.650, 0.549, 0.070, and 0.098 respectively. Similarly, sensor two has values of 0.700, 0.647, 0.075, and 0.137. OR logic will be examined with the number of looks at five. The DDDB with these values is displayed in figure 3.

Immediately, it can be seen that the results will not be satisfactory because the fused probabilities are just shy of the specifications and the non-decision probability has been driven down to 0 at five looks leaving no room for further gains. Hence, additional looks will not help. Likewise, fewer looks will degrade performance. Both of these conclusions can be seen by examining the "Plot Target" and "Plot Decoy" graphs. (See figures 4 and 5.)

One possibility, yet to be considered for these sensor inputs is the use of AND as opposed to OR logic. By examining the "Plot Target And/OR" and "Plot Decoy And/OR", it can be seen that AND logic shows increased fused correct probabilities for greater than five looks. The "Plot Decoy And/OR" graph is displayed in figure 6. By checking the AND logic on the DDDB, AND logic results can be more investigated further. An examination of the "Plot Decoy" graph, which can be found in figure 7, shows that a minimum of 6 looks will be needed, but unfortunately for 6 or more looks the fused probabilities for the target object do not meet the specifications. Hence, the basic sensor characteristics need to be tweaked. The probability of correct (T) decision will be increased to 0.6600.

With this new value, the "Plot Target And/OR" and "Plot Decoy And/OR" show that for five looks, OR logic is preferable but for 6 or more looks AND logic will provide better fused results. The “Plot Target And/OR" is shown in figure 8.

Unfortunately, an inspection of either the "Plot Target" or "Plot Decoy" (which is shown in figure 9) charts show that five looks will produce a probability of incorrect decision that is larger than acceptable. Hence, AND logic will be considered with six to eight looks. With six looks, the system requirements can be met.

Of course because this analysis is being done in the design phase, another useful fact is to know how many looks are required for fusion to be beneficial. An examination of the "Plot Target Fusion Benefits" and "Plot Decoy Fusion Benefits" shows that fusion benefits can be obtained in the range of three to eight looks. These plots are displayed in figures 10 and 11 respectively.

5. Concluding Comments

This work represents a continuing effort to further the application of fusion technologies by developing tools to aid in fusion system development. As a continuation of this effort, the same logic that was used to develop the theoretical foundations for determining fusion benefits for two sensors should be expanded to include three or more sensors and incorporated into FUSE. Furthermore, additional modules (similar to FUSE) covering other fusion modes should be added to increase the utility of the GIFTS architecture.

6. References


Figure 1: FUSE menu options

Figure 2: Data Definition Dialog Box with default values.
Figure 3: Initial analysis inputs

Figure 4: Plot of fused probabilities for the target object using initial inputs

Figure 5: Plot of fused probabilities for the decoy object using initial inputs
Figure 6: Comparison of probability of correct decision for decoy object using initial inputs under AND and OR logic

Figure 7: Plot of fused probabilities for decoy object using AND logic

Figure 8: Comparison of probability of correct decision for target object using revised inputs under AND and OR logic
Figure 9: Plot of fused probabilities for decoy object using OR logic and revised inputs

Figure 10: Fusion benefits for target object using revised inputs and AND logic

Figure 11: Fusion benefits for decoy object using revised inputs and AND logic